The Interaction of Word Recognition and Linguistic Processing in Speech Understanding[1]

H. Niemann, G. Sagerer[2], U. Ehrlich, E.G. Schukat-Talamazzini, F. Kummert

Lehrstuhl für Informatik 5 (Mustererkennung), Universität Erlangen-Nürnberg
8520 Erlangen, F.R. of Germany

Abstract: This contribution describes an approach to integrate a speech understanding and dialog system into a homogeneous architecture based on semantic networks. The definition of the network as well as its use in speech understanding is described briefly. A scoring function for word hypotheses meeting the requirements of a graph search algorithm is presented. The main steps of the linguistic analysis, i.e. syntax, semantics, and pragmatics, are described and their realization in the semantic network is shown. The processing steps alternating between data- and model-driven phases are outlined using an example sentence which demonstrates a tight interaction between word recognition and linguistic processing.

Keywords: speech understanding, semantic networks, scoring, linguistic analysis

1 Introduction

The main effort of automatic *speech recognition* is directed towards reliable, speaker-independent, and fast recognition of spoken sentences and words drawn from a sufficiently large vocabulary. Problems of linguistic analysis, understanding of the meaning, or derivation of an answer are of less interest. However, in many systems a *language model* is used to constrain the allowed word sequences. Examples of work in this direction are [4,5,6,13,20,25,26,28,32,38,42]. In *natural language* processing the main effort is towards linguistic analysis of printed texts, including text and story understanding as well as generation of answers to questions. Problems of falsely recognized or unrecognized words, incomplete sentences, or conversion of techniques developed for text understanding to speech understanding are of less interest. Examples of work in this direction are

[1] *Acknowledgment:* The work reported here has been supported by the German Ministry of Research and Technology (BMFT) in a 'Joint Research Project on Speech Understanding' and by the German Research Foundation (DFG) in the special program 'Models and Structural Analysis for the Evaluation of Image and Speech Signals'. This support is gratefully acknowledged; only the authors are responsible for the content of this paper.

[2] now AG Angewandte Informatik, Universität Bielefeld

NATO ASI Series, Vol. F 75
Speech Recognition and Understanding. Recent Advances
Edited by P. Laface and R. De Mori

[3,16,49,17]. The interest in speech is motivated (besides the pure scientific interest) in the facts that it provides an information channel which can be used independent of and in parallel to hand and eye, that it is a natural means of communication, that it can be used via standard telephone links, and that it allows a higher data rate in comparison to a keyboard. This allows interesting potential and actual applications [2,19,22,29], for example, dialog and language translation systems using speech for input and output.

A tacit assumption often is or has been that speech recognition sooner or later will achieve close to 100% recognition rate and that then the text input of a natural language system can just be replaced by the spoken words obtained from a speech recognition system. *Speech understanding* then consists of the two decoupled steps of word recognition and understanding of the meaning of an utterance. The decoupling implies that no interaction between the two steps is possible.

It has been argued frequently and supported experimentally that human speech understanding is a process incorporating *all* sources of evidence simultaneously. For example, the human recognition rate in phonetic transcription of meaningless words increases if the phonological rules of English are met. This supports the view that knowledge about speech on all levels of processing should be used [14], and in fact this is a standard approach in the automatic systems given in the references. It is supported from neurophysiolgy that certain subtasks are realized by independent modules [15]. This suggests a modular approach also to automatic systems. An early example of an automatic system having distinct modules (or knowledge sources) is HEARSAY II [11], and an early example of a system compiling all available knowledge into one network of states is HARPY [28]. Some systems for speech understanding are described in [18,27,23,31,47,52,51,53].

Of course, a *modular* system architecture *not* necessarily implies a *hierarchical* system. In such a system the i-th module accepts an input from the $(i-1)$-th module and passes an output to the $(i+1)$-th module. No information can be passed from module i to module $(i-1),(i-j)$, or $(i+j)$. A more flexible processing strategy should allow the system to focus on promising or important parts of the input at first, and then to inspect other parts in an order and amount of detail infered from intermediate results. This type of focusing or planning has been employed, for example, in [50]. In speech understanding the system might at first concentrate on semantically and/or pragmatically important words. An approach to implementing this type of flexibility is to use a modular system architecture with a distinct control module that determines the processing strategy. This means that the control module determines which processing module should be activated at which time using which data. In principle, it is well possible to activate p modules on p processors. This type of approach has been introduced and used, for example, in [11,34,33,36].

The purpose of this paper is to describe an approach to implement a speech understanding and dialog system which

- provides a well structured representation of general linguistic and special task specific knowledge such that
  - a homogeneous framework for knowledge representation on all levels of processing results,
  - the means for incorporating procedural knowledge of an arbitrary type are given;
- and simultaneously allows a flexible control strategy alternating between data-driven and model-driven phases of processing which includes
  - a theoretically based approach to achieve a global optimum of system behaviour,
  - the possibility to include local control heuristics if needed.

In particular, we consider the interaction between word recognition and linguistic processing which is enabled by the flexible control strategy. In this paper linguistic processing means syntactic, semantic, and pragmatic analysis. The goals of the system are described briefly in the next paragraph. The basis for system implementation is a homogeneous framework for knowledge-based speech understanding described in the next section. Sect. 3 concentrates on the judgment of word hypotheses and only gives a very short overview of word recognition, Sect. 4 describes linguistic processing up to the level of pragmatic interpretation of an utterance. The integration of these processing steps in a homogeneous system architecture is shown in Sect. 5, where also the processing strategy is presented. Results and a conclusion are given in Sect. 6 and 7, respectively.

The main goal of the system is to answer questions concerning a certain task domain using (German) speech for input and output. It has to perform the subtasks of *recognition* of words, *understanding* of the meaning of an utterance, generation of an *answer*, and if necessary generation of a *further inquiry* of the system. Derived from the four subtasks is the acronym EVAR. The vocabulary is between 1000 and 4000 words, the syntax accepts a reasonably large subset of German, word recognition is speaker-independent (but dialects are excluded), speech quality is limited to telephone bandwidth, the task domain is inquiries about German intercity trains, and the dialog module is in development [30]. Descriptions of the individual modules are available from [37,38,35]. This paper will provide recent approaches and results concerning the integration of modules.

2 An Approach to Knowledge Based Speech Understanding

Speech understanding is viewed as a sequence of operations transforming the speech input via different levels of abstraction to a desired output or also to an internal representation. The data structures for representation of the relevant knowledge and the results of operations are introduced, the task independent rules for using knowledge are defined, and an outline of a control algorithm which determines a processing strategy is given.

2.1 Data Structures

Representation and use of knowledge in a system for speech understanding is a primary problem. The knowledge consists of syntactic, semantic, pragmatic, and dialog knowledge, where the latter type is meant to allow the understanding of a sequence of utterances. In addition the relevant knowledge for the generation of an answer must be represented; as mentioned above, in the EVAR system this knowledge concerns intercity train connections. The required knowledge is represented in a *model* $\mathcal{M}$ which is a *semantic network*. The elements used here in the definition of a semantic network are illustrated in Fig.2.1. The network consists of

- three types of nodes:
  - the *concept*, which is a computer representation of a general definition or an *intensional definition* of a conception (e.g. of an object, event, fact, or meaning),
  - the *modified concept*, whose attribute values are more restricted (due to available results) than in the corresponding concept,
  - the *instance*, which corresponds to an actual occurrence of the corresponding concept in the sensor data,
- six types of links:
  - the *part link* relating a concept to one of its parts,
  - the *concrete link* relating the concept to one on a lower level of abstraction,
  - the *specialization link* relating a concept to a more specialized one,
  - the *reference link* relating a concept (e.g. the noun 'man') to one referencing it (e.g. the pronoun 'he'),
  - the *instance link* relating a concept and an instance of it,

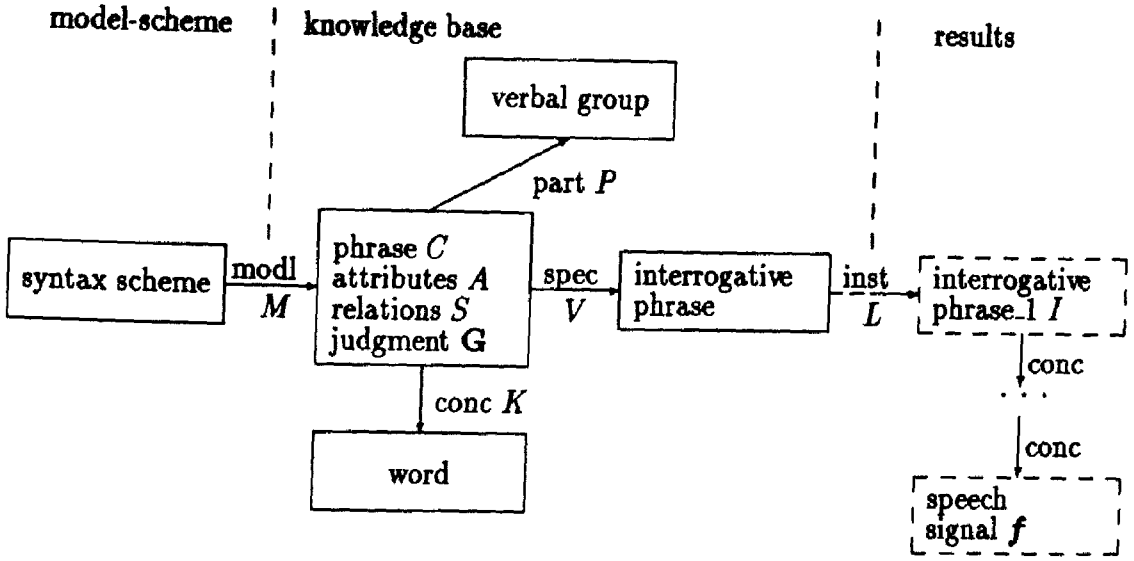


Figure 2.1. An illustration of a concept C and its links P, K, V, M, L to a part, concrete, specialization, scheme-concept, and instance, respectively

- the *model link*, relating a scheme-concept containing a priori knowledge to a concept derived from it (this is used for automatic knowledge acquisition and is not considered here),

- nine types of substructures which are used to define, for example, links, attributes, relations, or functions needed by the concept.

Hence, the only data type for knowledge representation is a concept

$$\begin{aligned} C &= \Big(D : T_C, (A : (T_A \mapsto F))^*, [H_{OBL}, H_{OPT}, H_{INH}]^*, (V : C)^*, \\ &\quad (R : C^+)^*, (M : C)^*, (L : I)^*, (S(A_C, A_P, A_K) \mapsto F)^*, (\mathbf{G} \mapsto F)\Big) \\ H &= \Big((P_{ci} : C^+)^*, (P_{cd} : C^+)^*, (K : C^+)^*\Big) \end{aligned} \tag{2.1}$$

An example is given in Fig.2.1. It has a name D (e.g. 'phrase'), a type T_C , attributes A (e.g. 'case, number, gender'), parts P (e.g. 'verbal group'), concretes K (e.g. 'words'), specializations V (e.g. 'interrogative phrase'), structural relations S (e.g. 'a noun must follow an adjective in time'), and a judgment vector G. Each attribute, structural relation, and judgment references a function F for the computation of a value.

A concept may have an arbitrary number of *sets of modality*. Each set is sufficient to define an instance of the concept and it consists in turn of an obligatory, optional, and inherent element denoted by H_{OBL}, H_{OPT}, and H_{INH}, respectively. Each element H may consist of an arbitrary number of context-independent and context-dependent parts as well as of concretes, denoted by P_{ci}, P_{cd}, K, respectively. As the notation implies, all

parts and concretes of an obligatory element must be available in order to instantiate the concept, and any subset of parts and concretes (including the empty subset) of an optional element is sufficient. An inherent element is one which is not observed in the sensory input, but which is implied by the meaning of the concept. For example, in EVAR a time table of train connections is inherent to a concept defining train connections. A context-independent part is one which can be instantiated without reference to an instance of the superior concept, a context-dependent part needs for its instantiation an instance of the superior concept.

A concept represents general knowledge which is used to analyse an utterance. Depending on the speech signal *instances* I of concepts C are computed. An instance ultimately relates a concept to an interval of the signal, possibly via several other instances. It has the same data structure as a concept, but references to functions F are replaced by computed *values*.

At a certain state of analysis of a speech signal some concepts will already have instances and for some concepts C_i it will still be impossible to compute instances because some prerequisites are missing. Nevertheless, the data available from the instances may allow one to impose additional constraints on the yet uninstantiated concepts. These more constrained concepts are called 'modified concepts'. For example, the available instances may request that the 'number' of an attribute may only be 'plural' or that the start-time of a noun group may only be 'behind' an already instantiated verbal group. The additional constraints are represented as restrictions on the set of values which can be computed by the functions F in a concept. These restrictions are defined in the above mentioned substructures.

2.2 Rules for Using Knowledge

In this subsection general rules are introduced to obtain an instance $I(C)$ of a concept C and to compute a modified concept $Q(C)$. These rules are *task-independent* in the sense that they can also be used, for example, in image understanding; task-dependent procedural knowledge is introduced by the functions F referenced by a concept.

There are three rules for computing an instance of a concept and three rules for computing a modified concept. Only the basic ideas are given here, a detailed definition of the rules is given in [34,40,44]. The six rules perform the following actions:

RULE_1 : create a partial instance ignoring context-dependent parts as well as optional parts and concretes; consider referential links during instantiation

RULE_2 : create an instance out of a partial instance taking into account context-dependent parts

RULE_3 : extend an instance by considering optional parts and concretes

RULE_4 : create a modified concept by bottom-up propagation of constraints

RULE_5 : create a modified concept by top-down propagation of constraints

RULE_6 : create a modified concept from attribute values or initial goal concepts

2.3 A Control Algorithm

Usually, at any state of analysis several out of the above defined six RULES will be applicable to several subsets of concepts, modified concepts, and instances. In order to achieve an efficient analysis focused to the goal of analysis it is mandatory to select the one or the few most promising alternatives. This makes up a processing strategy which is determined by a *control algorithm*. The general idea of such an algorithm is described in the following. It is based on an adaptation of the well-known A^*-graph search algorithm to the peculiarities of semantic networks [41,34]. The *advantages* of using this algorithm are that

- it provides a means for finding an optimal solution, hence enforces a clear definition of what is a 'good (intermediate) result',
- provides global control of the whole analysis process,
- but nevertheless allows one to introduce local and heuristic criteria via the procedures for the computation of the judgment attached to a concept.

It has been discussed in detail, for example, in [33,39,34], that the control problem may be viewed as the problem of finding an optimal path, the *solution path*, in an implicitly defined search graph $\mathcal{G}$ which is restricted here to a *search tree*. A node in this tree represents a state of analysis. It is important to distinguish the model $\mathcal{M}$ from the search tree $\mathcal{G}$. In an abstract sense both consist of nodes and links, but the *nodes* in $\mathcal{M}$ are *concepts*, whereas the *nodes* in $\mathcal{G}$ represent *states* of analysis. Hence, a goal concept in the model has to be distinguished from a goal node in the search tree. The goal concept may be known in advance, for example, it may be stated in advance that an answer to an uttered question is desired, that is the instantiation of an *ANSWER*-concept, or it may be stated that a pragmatic analysis of an utterance is desired, that is the instantiation of a *PRAGMATIC*-concept. The goal node in the search tree is *not known* in advance, rather its content will be the result of a successful analysis. A criterion for a goal node

may be, for example, that a good scoring instance of the goal concept has been computed and that this instance covers at least $p\%$ of the utterance.

The main steps of a control algorithm are summarized in Fig.2.2. In the context of semantic networks there are two types of transformations for generating successors of a node, the *expansion* and the *instantiation* of a model concept. Expansion of a model concept means that its successor concepts along the part and concrete links are generated and attached to the successor node; this corresponds to top-down or model-driven processing. Instantiation, of course, means the computation of an instance of a concept; this corresponds to bottom-up or data-driven processing. Hence, a flexible processing strategy *alternating* between model-driven and data-driven phases of processing is possible. Due to incomplete, noisy, and ambiguous data several alternative instances of one concept may be generated, for example, several instances of a noun group may result due to competing word hypotheses or ambiguous syntax. Every alternative instance is attached to a separate successor node in the search graph. Possible modifications of concepts are carried out during these steps.

The main assumption for determining a processing strategy is that a *judgment* ϕ can be computed for every node v in the search tree. The judgment ϕ of a search tree node should be distinguished from the judgment G of an instance of a concept. This point will be discussed below in more detail. According to the judgment it is possible to select from among the unexpanded nodes in the search tree the best scoring node v_k to be processed next. If a concept or modified concept in v_k can be instantiated, this is done first. If no instantiation is possible, it is tried to expand the node v_k along concrete and obligatory part links. If this is not possible, expansion along optional part and specialization links is tried.

2.4 Discussion

In the preceding subsections we presented a general framework for implementation of a knowledge-based speech understanding system. It is designed such that an *interaction* between the recognition and the understanding phases is possible. The main ideas are briefly summarized.

The system is structured into two main phases. The first one is a segmentation phase containing those processing steps which are executed mainly data-driven, in a fixed order, and without using task-specific linguistic knowledge. This is also called the speech front-end. The result is a set of word hypotheses which also may be checked syntactically by a grammar or a language model.

Input: application function to provide a list of goal concepts C_{gi}, $i = 1, \ldots, n$				
Initialize: application function and shell function to provide list *OPEN* with nodes v_{gi}, $i = 1, \ldots, n$ and their judgments $\phi(v_{gi})$				
WHILE *OPEN* is not empty DO:				
	select from the list *OPEN* the best scoring node v_k by application function *select*(*OPEN*) , remove v_k from *OPEN*			
	IF	application function *end_analysis*(v_k) decides that an analysis goal has been achieved		
	THEN	STOP - successful end of search or end of resource		
	IF	application function *goal_conc*(v_k) defines nonempty set S of new goal concepts		
	THEN	shell function *gen_goal*(v_k, S) generates new goals and corresponding nodes on *OPEN*		
	ELSE	IF	one object in v_k can be instantiated by one of the RULES_1,2	
		THEN	shell function *instant*(v_k) to instantiate v_k and to perform modifications by RULES_4,5	
		ELSE	IF	there is one object in v_k with an unfulfilled premise
			THEN	shell function *expand*(v_k) to expand v_k, to perform modifications by RULE_5, and to consider referential links
			ELSE	shell function *opt_spec*(v_k) to consider optional parts and specializations
STOP - unsuccessful end of search				

Figure 2.2. An outline of the main steps of a general control algorithm; it distinguishes actions which are task-independent and can be supplied in a system shell (shell functions) and actions which are task-dependent and have to be specified by the system designer (application functions)

The second phase is knowledge-based processing where a processing strategy is determined for every input by a control algorithm. The declarative knowledge is represented by a model $\mathcal{M}$ which is a semantic network of concepts C. Problem-specific procedural knowledge is referenced by functions attached to concepts. Speech understanding and dialog amounts to the computation of a description which is consistent with the model and the dialog context and optimally fits to the input speech signal. Instantiation of concepts is defined in a problem-independent manner by six RULES. A strategy for the instantiation of concepts is defined by a control algorithm which is a version of the A^*-algorithm adapted to semantic networks. Since it can alternate between data-driven and model-driven phases of processing it allows an interaction between word recognition and linguistic analysis. The task-independent part of this approach is implemented in the system shell ERNEST [40].

3 Scoring of Results from the Acoustic Front End

The main purpose of this section is to introduce a method for judging the quality of word hypotheses or also of chains of words generated by an acoustic front end. The judgment function should be in accordance with the requirements of subsequent knowledge-based processing. It is not intended here to describe the acoustic front end in detail; details are given in [24,43,46].

3.1 Overview

In the acoustic front end the speech signal f is processed to obtain a lattice of word hypotheses. The first processing step is a parametric representation of the speech signal which in EVAR is based on smoothed delta Bark spectrum Lem cepstrum coefficients. In addition the parametric representation includes suprasegmental features. A segmentation into phonemic subword units is carried out next. Based on a lexicon, the structure of which is discussed in detail in Subsect.4.1, a set of word hypotheses is determined. A word is represented by a hidden Markov model (HMM). In addition to the lexicon syntactic constraints may be used to reduce the number of word hypotheses. These processing steps make up the *acoustic front end*. The result is a set of words or of syntactically correct word chains.

3.2 Judgment of Acoustic Quality

The *judgment of a hypothesis* for a word or a word chain should meet the conditions of the A*-algorithm for graph search (see for example [41,34]), since it is used to control linguistic processing. The search must be *admissible* in the sense that the best sequence of words is found even if not all sequences are evaluated. This can be achieved if the *judgment of a node* v_i in the search space is estimated by

$$\hat{\phi}_q = \hat{\psi} + \hat{\chi}, \tag{3.1}$$

where

- $\hat{\psi}$ is an estimate of the cost of a path from the start node to v_i
- $\hat{\chi}$ is an estimate of the cost of a path from v_i to a goal node
  (this path is not yet known if the search proceeded only to v_i)

In order to achieve an *admissible algorithm*, the estimate $\hat{\chi}$ must be *optimistic*, that is it must be *smaller* than the true costs χ.

For the word hypotheses this means that the judgment should evaluate the cost of that part of the utterance *covered* by the hypothesis ($\hat{\psi}$) plus the cost of the part *not covered* by the hypothesis ($\hat{\chi}$). The term $\hat{\psi}$ is based on the score obtained from word recognition. For $\hat{\chi}$ we use an estimate of the rest which is optimistic in most cases.

The term $\hat{\psi}$ is based on (but not necessarily identical to) the score G_q obtained from *word recognition*, that is in DTW the sum of local distances and in HMM the product of pairs of probabilities for state transition and output, or the sum of their (negative) logarithms, that is $G_q = -\log[p(o_b \ldots o_e \mid HMM)]$. Hence, a simple *'linear model'* of the quality is

$$G_q = \sum_{\tau=1}^{T} G_\tau , \tag{3.2}$$

where T is the length (in frames of speech) of the word hypothesis.

Let us assume *independent identically distributed* random variables G_τ with mean μ_c and variance σ_c^2 for a *correct hypothesis* and mean μ_f and variance σ_f^2 for a *false hypothesis.* In this case the cost of a hypothesis of length T has a distribution with

- $(T\mu_c, T\sigma_c^2)$ for a correct hypothesis,
- $(T\mu_f, T\sigma_f^2)$ for a false hypothesis.

It has been *verified experimentally* that mean and variance of the quality of correct and false hypotheses depend linearly on T as predicted by the above 'linear model' for the judgment of hypotheses [45].

During *word hypothesization* the set of N best scoring hypotheses is selected. The *first step of linguistic processing* is to select from the set of word hypotheses the N' best scoring pragmatically relevant hypotheses. The *subsequent steps of linguistic processing* generate linguistically meaningful sequences of words, for example, syntactic constituents, case frames of verbs, instances of pragmatic concepts (e.g. an instance of the concept *TRAIN_CONNECTION*), or instances of dialog concepts. Each such word sequence

- causes the generation of a node v in the search tree,
- is judged by a vector **G**, one of whose components measures the *acoustic quality* G_q of n words in a sequence,
- is used to estimate the quality $\hat{\phi}_q(v)$ of the corresponding search tree node.

Let $w1, w2, \ldots, wn$ be the words in the word sequence. The acoustic quality is defined

by

$$G_q = \sum_{i=1}^{n} G_{qi} = \sum_{i=1}^{n} -\log[p(o_{bi} \ldots o_{ei} \mid HMM_{wi}])$$ (3.3)

This is also a good estimate for non-adjacent words.

The *quality of the search tree node* is in accordance with the requirements of the A*-algorithm defined to be

$$\begin{aligned} \hat{\phi}_q &= G_q + G_r \\ &= \hat{\psi} + \hat{\chi} \end{aligned}$$ (3.4)

An approach to determine G_r is from the 'linear model' of scoring. With some constant γ we define

$$G_r = T_u \mu_c - \gamma \sigma_c \sqrt{T_u},$$ (3.5)

where T_u is the duration of that part of the utterance *not covered* by the word sequence. This ensures an optimistic estimate $\hat{\chi}$ in a certain percentage of cases determined by the constant γ. For example, if the distribution of quality judgments G_q of a word of length T were normal (which is at best an idealization), $\gamma = 2$ would ensure that at most 3% of hypotheses of length T would have a better quality score.

This estimate of χ is better than the 'optimistic' estimate $\hat{\chi} = 0$ and according to our experiments it is also better than the 'shortfall score' [50].

4 Linguistic Constraints for Speech Understanding

This section describes the linguistic constraints used in EVAR. The constraints range from recognized words or word chains through the levels of syntactic, semantic, and pragmatic analysis up to carrying out a man–machine dialog. The so–called 'raw linguistic knowledge base' is the lexicon which is used by various levels in the system; for each level a preprocessor extracts the relevant information from the lexicon. Syntactic processing as understood here determines the syntactic structure of an utterance taking into account only rules for the combination of (syntactic) word classes without considering their meaning. Semantic processing determines the meaning of words, but only considers general task–independent semantic properties of words. Pragmatic processing interprets *one* utterance with respect to a model of the task domain. Finally, dialog processing — among others — interprets one utterance in the *context* of preceding utterances. This allows a layered or stratified approach to the representation of declarative knowledge and facilitates changes in the linguistic competence of one level without affecting others. Since according to Subsect.2.3

a distinct control module is used, the stratified approach nevertheless allows a flexible processing strategy.

4.1 The Lexicon

Since detailed descriptions of the lexicon and justifications of the design choices are available from [9,10,35], only a short overview of the main points is given in the following.

The unit of the lexicon, i.e. an individual entry in the lexicon, is defined by the spelling of a word or by the *graphematic word*. A unit of the lexicon is simply called a *word*. Every word has associated with it a unique integer number for its identification. The information attached to a word consists of the tuple [spelling, word number, pronunciation, syntactic information (may be multiple per word), semantic information (may be multiple per syntactic definition) and possibly dialog attributes, pragmatic information (may be multiple per semantic definition)]. A word may have only a part of this information. The detailed format of a lexicon entry is specified in [10,35]. Each alternative for the syntactic, semantic, or pragmatic definition is called a syntactic, semantic, or pragmatic word, respectively. In order to facilitate word recognition it was decided to include all inflections of a word as separate entries.

Presently the lexicon has 4427 words whose entries at least contain [spelling, word number, pronunciation]. They are used for word recognition experiments with vocabularies of different sizes. The lexicon contains 3953 syntactic words, out of which 841 are base forms, 1124 semantic words, or lexemes, and 246 pragmatic words in the above defined sense. The syntactic, semantic, and pragmatic information is described in the corresponding subsections. A set of preprocessors has been implemented which extracts the information necessary for a certain processing algorithm and transforms it to a useful format. The result is called a sub–lexicon. Since all sub–lexica are derived from the same lexicon, consistency of sub–lexica is guaranteed. For example, syntactic processing only needs word number and syntactic information. This approach separates acquisition and maintenance of the lexicon from the design and implementation of processing modules.

The lexicon is implemented in LISP as a database which in particular has an editor and search functions. The editor supports the addition of new words and the modification of existing words; it also checks the formal consistency of entries.

4.2 Syntactic Constraints

Syntactic analysis is based on constraints defined by an augmented transition network (ATN) grammar of a subset of German language. Only *simple syntactic constituents* are

determined, but not complete sentences. By a 'simple' syntactic constituent we mean a group of words containing only one nucleus, for example, 'the next train', but not 'the next train to Hamburg'; the latter group would be treated as two constituents. This approach has the following reasons. In a spoken dialog it often occurs that not a complete sentence but only a fragment is uttered which usually is fully understandable in the context of a dialog. Since word recognition is not perfect, some words in a sentence may be missing, but many of the constituents may still be recognized correctly. Due to missing correct words and wrongly recognized words a large number of wrong sentences which are accepted by the ATN may be generated. The separation of syntax, semantics, and pragmatics is useful for a modular representation of knowledge. As mentioned already, this does not prevent a flexible processing strategy. The syntactic constituents are checked immediately for semantic consistency as described in the next subsection. This is done because syntax alone imposes only weak constraints on a constituent, in particular if a grammar for a sufficiently large subset of German is used. Finally, if a syntactic constituent is pragmatically relevant, it can be tried at a very early stage of processing to verify the corresponding pragmatic concept top–down or model–based. For these reasons the so–called 'constituent grammar' does not specify sentences.

Usually, a syntactic constituent is coherent in the sense that the words belonging to it follow each other in time. An exception are verbal groups VG which in German may be non-coherent. Therefore, verbal groups are not checked initially.

The grammar only contains constructions occurring in inquiry dialogs. Therefore, it does not contain passive forms. Meta–communicative forms, for example, 'thank you' or 'good morning', are treated by the dialog module. Ideally, a grammar should be task–independent. However, since it accepts only a subset of German, this subset is chosen to meet the requirements of a task–domain. For example, the syntactic constituent 'DATUM' (date) is important for inquiries about intercity trains, but it may be unimportant and omitted for inquiries about traffic rules.

Syntactic analysis using the full ATN is performed under the control of the algorithm described in Subsect.2.3. Depending on the judgment of search tree nodes syntactic analysis may proceed data–driven or model–driven, and it may switch between those two phases. Hence, there is no distinct parser. But it is emphasized that a special parsing algorithm could be integrated into the semantic network approach if desired. This can be done, for example, in one of the following two ways: first, by placing the interface between the semantic network and the other processing steps not at the level of words but at the level of syntactically parsed word strings; second, by leaving the interface at the level of words and attaching the parser as a procedure to one concept *SYNTAX*.

4.3 Semantic Constraints

The general task independent meaning of words and utterances is represented on the level of *semantics*. It is based on case and valency theory as developed in [1,12,48], on a system of semantic classes of words, and on constraints between semantic classes of words within a syntactic constituent. Semantic analysis consists of three steps. First, the semantic consistency of simple syntactic constituents generated according to the ATN grammar is checked. Second, simple syntactic constituents are combined to complex syntactic constituents. Examples are constituents like 'the next train to Hamburg' or 'on Monday morning at about 8 o'clock'. Third, the constituents are summarized to a sentence hypothesis including also verbal groups.

Valency theory is outlined shortly using the verb valency as an example. In EVAR a noun, adjective, and adverbial valency is defined in a similar way as well. A verb used in a particular meaning requires a certain set of obligatory and optional elements having well-defined syntactic and semantic properties. These elements define the 'valency frame' of a verb in a particular meaning. If the same verb (the same graphematic word in the sense of Subsec.4.1) is used in another meaning, it will require a different set of elements thus defining a different valency frame. The above mentioned obligatory elements are those necessary to obtain a grammatically correct sentence. The optional elements are those which in addition to the obligatory ones are necessary to fully define the verb meaning. In German it is possible to add an almost arbitrary number of so–called 'free elements' which can be used fairly independent of a certain verb meaning. Therefore, free elements are not checked in order to determine the meaning of a verb, but they may have to be considered in order to determine an interpretation spanning a complete utterance. The emphasis of valency theory is on the syntactic structure of a verb and on selectional semantic restrictions between words.

Case theory concentrates on the 'functional role' or the 'deep case' of a syntactic constituent in a sentence. For example, in the sentence 'Tom washes the car' and 'the car is washed by Tom', the deep case of 'Tom' as an AGENT performing a certain action is the same although the syntactic properties are different. Thus the logical relations of a word with respect to a verb are considered in case theory. In EVAR 29 deep cases are distinguished which can be found in [35]. Examples are the GOAL, PATH, SOURCE, and TIME of a journey.

According to case and valency theory the *case frame* of a verb (or a noun, adjective, adverb) is determined. The relevant raw semantic knowledge is contained in the lexicon and extracted by a preprocessor. The elements of this knowledge depend on the word class and are given in Fig.4.1.

word class		semantic knowledge	
ADJ	attributive adjective	selection class	case frame
ADV	adverb	semantic class	relation
DET	determiner	negation or not	reference
N	noun	semantic class	case frame
PRAEP	preposition	semantic class	selection class
VERB	verb	semantic class	case frame

Figure 4.1. Types of semantic knowledge associated with a certain syntactic word class

Semantic classes of words are selected from a set of 107 general task-independent meanings used in EVAR. Possible meanings of a class of words are specified in a tree structure of semantic classes. For example, the syntactic word class VERB may have the semantic classes Act (action, e.g. drive a car), Chg (change of state, e.g. drive to Hamburg), Pro (process, e.g. to fall down), Stt (state, e.g. to sit), and Tru (truth value, e.g. could be). A semantic class may have subclasses. For example, the class 'Chg' mentioned above has the subclasses Mov (movement, e.g. to drive), Per (perception, e.g. to see), and Cmm (communication, e.g. to speak).

According to Fig.4.1 a noun has a 'semantic class' and a 'case frame', a preposition has a 'semantic class' and a 'selection class'. For the combination of a preposition (or an adjective) with a noun the selection class and the semantic class are used to represent restrictions on the combination of words used in a particular meaning. The restriction is defined by the rule

IF (a preposition or an adjective is used in the meaning defined by its semantic class),

THEN (the associated noun must have a semantic class corresponding to the 'selection class' of the preposition). (4.1)

An example is given in Fig.4.2. It shows a noun and a preposition which both have two possible meanings resulting in $2 \times 2 = 4$ possible combinations for the prepositional noun group 'in the coach'. But only the selection class Location of the preposition occurs as semantic class of one meaning of the noun. Hence, only one combination of meanings is selected in this case.

In summary, semantic analysis consists of the three steps:

1. Checking of simple syntactic constituents for semantic consistency. This is the *filtering of syntactic constituents*. It is mentioned that in order to increase the efficiency of this step the pragmatic consistency of constituents is checked, too. There is a test for consistency of adjectives and prepositions with the corresponding noun (thus

noun: coach	
meaning_1: railway carriage	meaning_2: trainer in athletics
semantic class: Transport, Location	semantic class: Acting_person
preposition: in	
meaning_1: in the evening	meaning_2: in the room
semantic class: Duration selection class: Time	semantic class: Place selection class: Location

Figure 4.2. Example of a noun and a preposition both having two possible meanings; note that the 'meaning_i'-entry ($i = 1, 2$) is only for readability of the lexicon

eliminating combinations like 'the fast tree' or 'inside the morning'). It is tested for nouns used in the singular number and having no article (thus eliminating combinations like 'how much is ticket'). The contradiction of semantic features is tested (thus eliminating cases like 'take a next train'). Finally, there is a test for constituents in the genitive and depending not on a noun but on a verb or an adjective, since the latter two are rarely used in inquiry type dialogs; however, they are reconsiderd when searching for complex constituents.

2. The *construction of complex constituents.* It consists of the construction of genitive constituents (for example, 'the departure of the train'), the construction of denominal actants (for example, 'the train to Heidelberg' or 'ten kilometers away'), the identification of anaphoric references requiring a dialog memory, the generation of sentence hypotheses for one sentence with a verb (for example, 'I want to go to Hamburg') using valency and deep case information from the lexicon, but not using all available restrictions in this phase, and finally the construction of temporal constituents (for example, 'tomorrow in the morning at about 8 o'clock').

3. The instantiation of a sentence hypothesis in a semantic network representing the full semantic knowledge. It consists of the two phases of the structural interpretation of a sentence hypothesis in the semantic network including *all* available restrictions, and the inclusion of free elements if there are any.

It is emphasized that the above steps *need not be executed sequentially* since the semantic network environment provides a very flexible control algorithm. Pragmatic interpretation is discussed briefly in the next Section, in particular Fig.5.3.

5 System Integration

In the preceding Sects.2-4 we presented a general approach to knowledge-based speech understanding, the scoring of results from the acoustic front end, and the main steps of linguistic analysis. This section shows the *integration in a homogeneous system* based on semantic networks. An important point is that a semantic network allows a *well structured representation of knowledge* maintaining the idea of a stratified system and simultaneously facilitates a *flexible use of knowledge* thus enabling an alternation between data-driven and model-driven processing steps as well as immediate transmission of results and requests via several levels of representation. The network implementation of linguistic conceptions is discussed, the handling of model-based requests passed from the linguistic analysis to the acoustic front end, the processing strategy, and the judgment of instances and search tree nodes.

5.1 General Ideas

The basic ideas of integrating a complete speech understanding system into a semantic network, for example, into the ERNEST environment are:

- map declarative knowledge to concepts, that is in particular knowledge about syntax, semantics, pragmatics, dialog, task domain (Intercity train connections in EVAR), and answer generation;
- attach procedural knowledge to concepts (e.g. for computation of attributes, relations, judgment);
- provide a judgment function meeting the requirements of the A*-algorithm for graph search;
- provide the application-functions for the control algorithm (or otherwise be content with the default functions provided in ERNEST);
- provide a (software) interface between acoustic front end and linguistic processing.

An overview of the network is given in Fig.5.1.

5.2 Representation of Linguistic Knowledge

In this subsection we consider the representation of syntax with the example of a noun group and the representation of a section of the task domain in ERNEST.

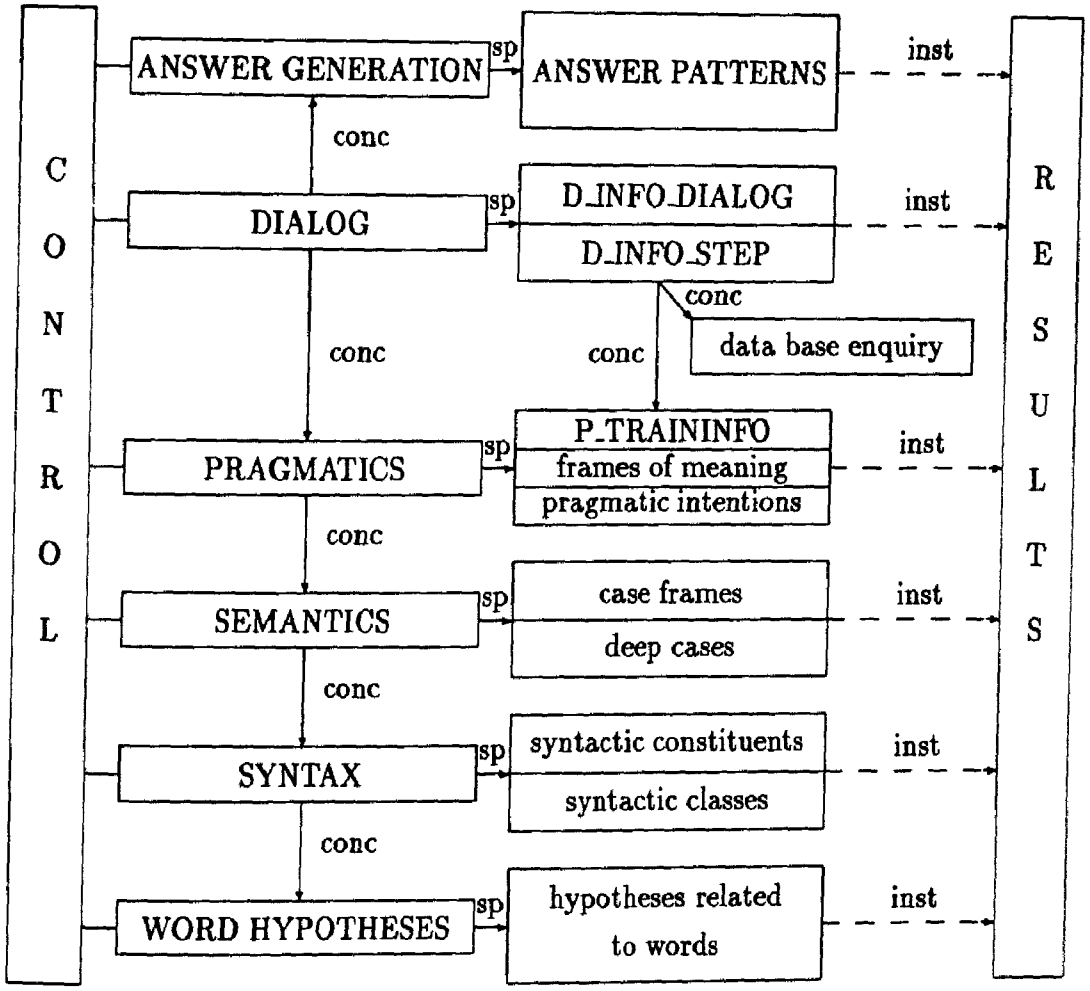
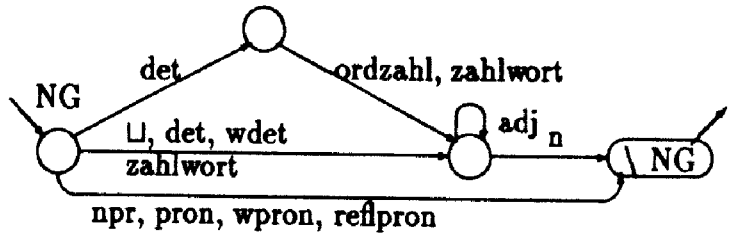
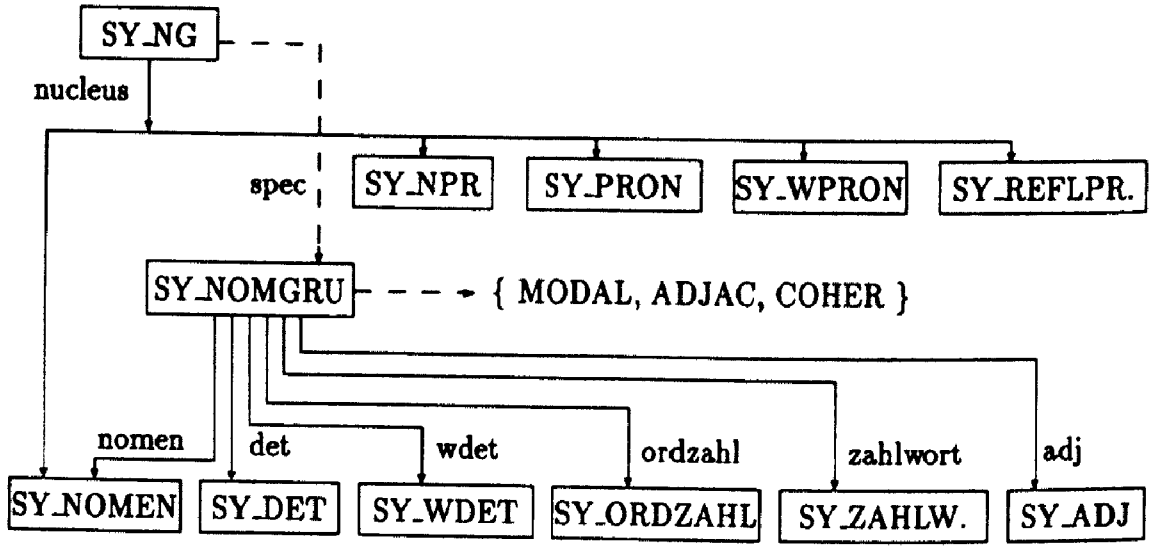


Figure 5.1. An overview of the semantic network representing the speech understanding system

Fig.5.2 shows a noun group in the common RTN representation and as concepts in a semantic network. The other syntactic constituents of the constituent grammar are represented in a similar way. The elements occurring on the edges of the RTN are represented by concepts (rectangles in the figure), the alternative definitions of a noun group correspond to different sets of modality (H_{OBL} and H_{OPT} in the figure), and the time order of elements is represented by an adjacency matrix (ADJAC in the figure).

Next we consider a section of the task domain as shown in Fig.5.3, where concepts are denoted by P_(name) on the pragmatics level, S_(name) on the semantics level, and SY_(name) on the syntax level. It defines on the pragmatics level the obligatory and optional elements of an inquiry for a train connection (*P_TIME_TABLE*). Since this model defines the elements necessary for such an inquiry, it is possible for the dialog module to generate a request for further particulars if one of the obligatory elements is missing





MODAL : H_{OBL} : { nomen }
H_{OPT} : { det, wdet, ordzahl, zahlwort, adj }

ADJAC :

	nomen	det	wdet	ordz	zahl	adj	
nomen	0	1	1	1	1	1	
det	0	0	0	0	0	0	
wdet	0	0	0	0	0	0	
ordzahl	0	1	0	0	0	0	
zahlwort	0	1	0	0	0	0	
adj	0	1	1	1	1	2	

⇒ a word class in column may precede a word class in row

COHER : YES

⇒ constituents must be adjacent in time

Figure 5.2. An RTN of a noun group and its representation by a semantic network

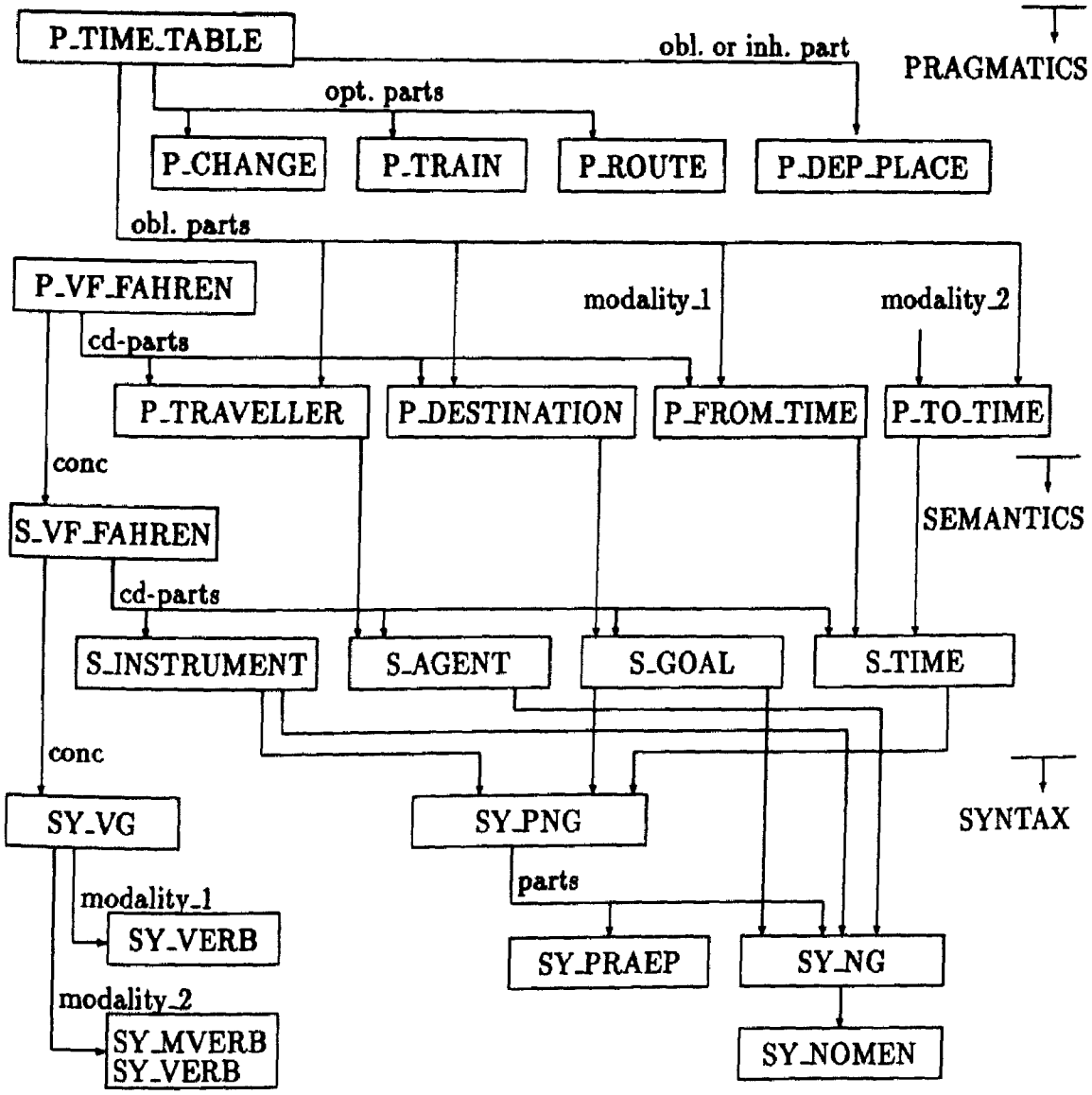


Figure 5.3. An overview of the concepts in a *section* of the model of the task domain; not all concepts and not all links are shown

(e.g. because it was not recognized or not uttered). Among the parts is, for example, a concept *P_DESTINATION* which in turn is referenced as a context-dependent part by the verb frame *P_VF_FAHREN*. Hence, *P_DESTINATION* can only be instantiated if a partial instance of *P_VF_FAHREN* is available. Of course, there are other verb frames (and also noun frames) referencing *P_DESTINATION* such that different alternatives for its instantiation are possible. A concrete of *P_VF_FAHREN* is on the semantics level *S_VF_FAHREN* and on the syntax level *SY_VG* for a verbal group. It is seen that, for example, *P_VF_FAHREN* has a part *S_TIME* which in the pragmatic context is specialized to either *P_FROM_TIME* (departure time) or *P_TO_TIME* (arrival time).

5.3 Judgment

Two types of judgment have to be distinguished:

1. The judgment **G** of modified concepts $Q(C)$ and instances $I(C)$ of a concept C.
2. The judgment ϕ of a node v in the search tree.

   The node contains (among others)

   - a goal concept C_g from the model which has been modified to $Q(C_g)$ or instantiated to $I(C_g)$,
   - the modified concepts and instances generated so far in order to modify and instantiate C_g,
   - potentially the whole model and all intermediate results.

The judgment of modified concepts and instances is defined by functions referenced by the corresponding concept.

The estimate $\hat{\phi}$ of the judgment of a search tree node is defined to be the current judgment of the modified or instantiated *goal concept* C_g, subject to the requirements of the A*-algorithm.

The judgment **G** of an instance I or a modified concept Q of a concept C is the vector $\mathbf{G} = (G_c, G_q)_t$. The components are

- G_c: the compatibility of a hypothesis with the linguistic knowledge (a binary number),
- G_q: the quality of a word sequence (that is, not necessarily adjacent words) making up the hypothesis with respect to the speech signal — in the present implementation *without* the estimate of the remainder according to the A*-algorithm.

Since G_c is a binary number we do not use a stochastic language model presently. The reason is that a stochastic model needs a large sample for the estimation of its parameters and that it is adapted to the average statistics of this sample, but not to the statistics occurring in a particular dialog situation. Meanwhile significant progress has been made in estimating the statistics of speech and in statistical modeling of dialog situations [21,8]. If an appropriate statistical model is available, it can be used in the above approach by modifying G_c, G_q appropriately.

The judgment of a search tree node is based on the idea that

- the computed result should be compatible with the linguistic knowledge and the dialog context,

- and it should have maximal 'similarity' to the speech signal.

The judgment $\hat{\phi}$ of a search tree node v having the associated goal concept C_g with current modification $Q(C_g)$ is the vector

$$\hat{\phi} = (\phi_c, \phi_q, \phi_r, \phi_t, \phi_p)_t. \tag{5.1}$$

It is evaluated in lexicographical order using intervals for some components.

The components are

- ϕ_c: the compatibility of a hypothesis with the linguistic knowledge

  $$\phi_c = G_c(Q(C_g)) \in \{0, 1\},$$

  since first of all, a hypothesis must be compatible with the linguistic knowledge;

- ϕ_q: the quality of the word chain making up the hypothesis with respect to the speech signal according to (3.4)

  $$\phi_q = G_q(Q(C_g)) + G_r(Q(C_g)),$$

  since among the compatible hypotheses we prefer the one having best quality;

  the above two components ensure the computation of compatible hypotheses having optimal acoustic quality; the following components are used to further reduce the amount of search:

- ϕ_r: the reliability of the hypothesis

  $$\phi_r = [\text{number of speech frames in the longest word chain (adjacent words)}],$$

  since among those having high quality we prefer the most reliable one (long words are recognized more reliably than short ones);

- ϕ_t: the total coverage of the utterance

  $$\phi_t = [\text{number of speech frames covered by word hypotheses in } Q(C_g)],$$

  since among the reliable hypotheses we prefer those best covering the utterance (i.e. yielding a final result more quickly);

- ϕ_p: the pragmatic relevance of the hypothesis

  $$\phi_p = G_p(Q(C_g)),$$

  since finally, we prefer the pragmatically most relevant one. The definition of pragmatic relevance uses fuzzy functions [7] and details are given in [9].

6 Processing Steps and Results

In the following we consider as an example some processing steps of the sentence

Wir möchten am Wochenende nach Mainz fahren

(We would like to go to Mainz at the weekend)

After initial generation of 100 word hypotheses processing on a RISC work station (about 14 mips, 6MB) took 101 sec CPU time until the instantiation of the (correct) instance of the pragmatic concept *P_TIME_TABLE*. During the analysis the control algorithm generated 1053 search tree nodes, 1233 modified concepts, and 777 instances.

The following are the test conditions:

1. Sentences in continuous speech sampled at 10 kHz using telephone bandwidth.
2. Word recognition and word verification are speaker independent.
3. The lexicon for recognition and verification has 1250 entries.
4. The judgment vector is as described previously.
5. Tests were performed with 100 word hypotheses including *all* words actually spoken.

So far only 12 sentences were analysed, but a larger sample is in preparation.

Linguistic processing is initialized using as initial goal concepts C_{gi} the following concepts from the syntax level:

- *SY_ADJ*, *SY_ADJU*, *SY_ADV*, *SY_NOMEN*, *SY_NPR*, *SY_ORDZAHL*, *SY_PRON*, *SY_ZAHLWORT*, *SY_WADV*

The control algorithm initializes a search tree node v_{gi} for each goal concept. From the 100 word hypotheses the 10 best scoring pragmatically relevant ones are used to instantiate the above goal concepts.

Next the control algorithm provides the option to generate a list S of new goal concepts as evident from Fig.2.2. This option is implemented presently in a 'look-up table' providing as the list S the following concepts on the pragmatics level:

$$S = \{P\_TRAVELLER, P\_DEP\_PLACE, P\_ROUTE, P\_DESTIN \; P\_TRAIN, P\_CHANGE, P\_FROM\_TIME, P\_TO\_TIME\} \quad (6.1)$$

In the model — a section of which is shown in Fig.5.3 — it is now tried to find a path along 'concrete-of' or 'part-of' links from an instantiated concept (e.g. *SY_NOMEN*) to

one of the new goal concepts (e.g. *P_FROM_TIME*). In order to increase the efficiency of this step

- only concepts on the path are considered having a pragmatic class compatible to that of the instance of the starting concept,
- modifications and instantiations possible on this path are carried out.

After this bottom-up step the verification of the pragmatic concept *P_FROM_TIME* follows as a top-down step.

Next the case frame of a verb is tried since a verb (or a noun) references pragmatic concepts as context-dependent parts in its case frame. In general, in a pragmatic concept (like *P_FROM_TIME*) *all* possible contexts (verbs, nouns) are referenced, but only one example is shown in Fig.5.3. For every context a modified concept is created by RULE_4 and a new search tree node is generated. By iterative application of RULE_5 the model is expanded, in this case for the verb frame.

The control algorithm allows in principle an arbitrary number of steps for the generation of new goal concepts. If the verb frame is instantiated, new goal concepts on the level of inquiries are generated. For the concept *P_FROM_TIME* this gives

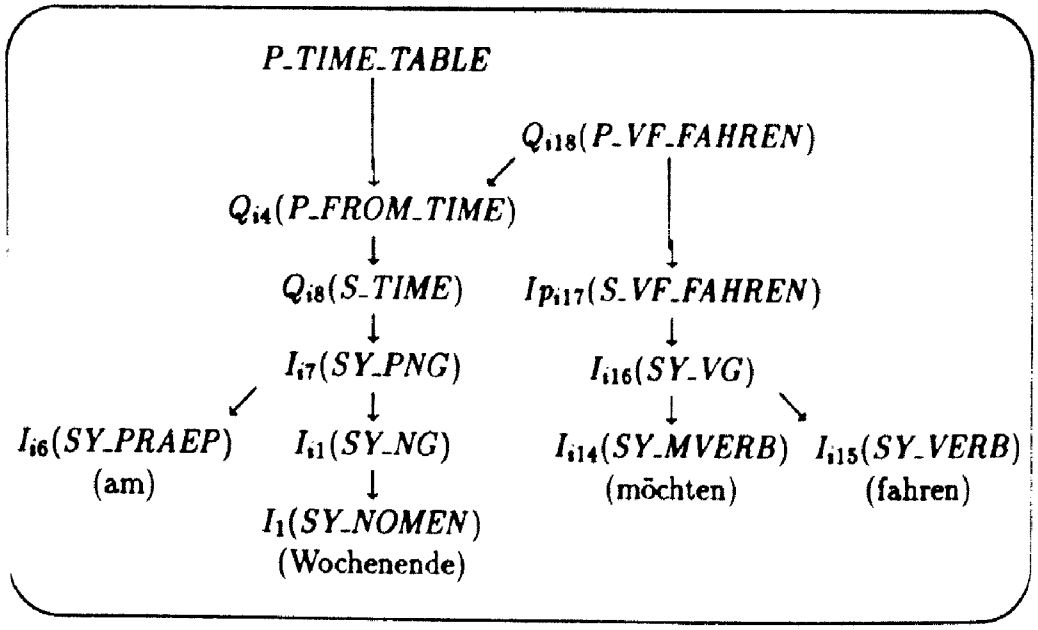
- *P_CONNECT_INFO*, *P_TIME_TABLE*

As an example the content of a search tree node representing this intermediate state of analysis is shown in Fig.6.1. It is seen that it contains the complete structure of the utterance analysed so far.

Finally the goal concept on the pragmatics level is instantiated. The instance has to meet the conditions for a successful analysis:

1. instantiation of a top-level concept,
2. coverage of at least 80% of the utterance.

This way control alters between expansion of the model and modification of expanded concepts (by RULE_5) and instantiation of the model (by RULES_1,2,3) and thus achieves an interaction between word recognition and linguistic processing.



$$\phi = (1, -4818, 77, 97, 2)_t$$

Figure 6.1. A search tree node showing an intermediate state of analysis of an utterance and its judgment vector

7 Conclusions

The main points of this contribution are:

1. *Knowledge representation and use* in a general system shell based on semantic networks and a flexible control algorithm allowing well-structured knowledge representation as well as alternating phases of data-driven and model-driven processing.

2. *A system integration* of a speech understanding system from the level of word hypotheses to the pragmatics level interfacing the acoustic front end to the linguistic knowledge base.

3. *Results* for the understanding of several sample sentences, illustrated by one particular sentence, and implemented on a standard hardware using standard software.

Our present and future work is directed towards improved word recognition, extended linguistic competence including a larger vocabulary, integration of dialog and answer generation, improving the efficiency of control and hence of processing, coping with utterances containing more than one sentence, and integrating prosodic cues to word recognition and linguistic analysis.

References

[1] W. Abraham, editor. *Valence, Semantic Case, and Grammatical Relations, Vol. 1.* John Benjamins, Amsterdam, 1978.

[2] W.A. Ainsworth. *Speech Recognition by Machine.* Volume 12 of *IEE Computing Series*, Peter Peregrinus Ltd., London, 1988.

[3] J. Allgayer, K. Harbusch, A. Kobsa, C. Reddig, N. Reithinger, and D. Schmauks. XTRA: a natural language access system to expert systems. *Int. Journ. of Man Machine Studies*, 31:161–195, 1989.

[4] L.R. Bahl, F. Jelinek, and L.R. Mercer. A maximum likelihood approach to speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5:179–190, 1983.

[5] J.M. Baker. Dragon dictate TM-30K: A natural language speech recognition with 30,000 words. In *Proc. European Conf. on Speech Communication and Technology*, pages 161–163, Paris, 1989.

[6] J.S. Bridle. Alphanets: a recurrent 'neural' network architecture with a hidden Markov model interpretation. *Speech Communication; special issue on Neurospeech*, 6, 1989.

[7] R. DeMori. *Computer Models of Speech Using Fuzzy Algorithms.* Plenum Press, New York, 1983.

[8] R. DeMori, J. Bourdeau, and R. Kuhn. A probabilistic approach to person-robot dialogue. In P. Laface and R. De Mori, editors, *Recent Advances on Speech and Language Modeling, NATO ASI Series F*, Springer, Berlin, Heidelberg, 1990.

[9] U. Ehrlich. *Bedeutungsanalyse in einem sprachverstehenden System unter Berücksichtigung pragmatischer Faktoren. Sprache und Information Bd. 22*, Max Niemeyer, Tübingen, 1990.

[10] U. Ehrlich. *Ein Lexikon für das natürlich-sprachliche Dialogsystem EVAR.* Volume 19 of *Arbeitsberichte des Inst. für Mathematische Maschinen und Datenverarbeitung*, Universität Erlangen-Nürnberg, Erlangen, F. R. of Germany, 1986.

[11] R.D. Fennell and V.R. Lesser. Parallelism in artificial intelligence problem solving, a case study of HEARSAY II. *IEEE Trans. Computers*, 26:98–111, 1977.

[12] C. Fillmore. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguisitic Theory*, pages 1–90, Holt, Rinehardt, and Winston, New York, 1968.

[13] L. Fissore, P. Laface, G. Micca, and R. Pieraccini. A word hypothesizer for a large vocabulary continuous speech understanding system. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 453–456, 1989.

[14] H. Fujisaki, K. Hirose, H. Udegawa, and N. Kanedera. A new approach to continuous speech recognition based on considerations on human process of speech perception. In *Proc Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 1995–1962, Tokyo, 1986.

[15] N. Geschwind. Specializations of the human brain. *Scient. American*, 241:158–168, No. 3 1979.

[16] B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira. TEAM: An experiment in the design of transportable natural language interfaces. *Artificial Intelligence*, 32:173–244, 1987.

[17] L. Hirschman, F.-M. Lang, J. Dowding, and C. Weir. Porting PUNDIT to the resource management domain. In *Speech and Natural Language Workshop*, pages 277–282, Philadelphia, 1989.

[18] H. Hoege and H. Ney. Das Projekt SPICOS: Organisation und Systemarchitektur. *Kleinheubacher Berichte*, 29:29–36, 1986.

[19] J. Hollingum and G. Cassford. *Speech Technology at Work.* Springer, Berlin, Heidelberg, 1988.

[20] F. Jelinek. The development of an experimental discrete dictation recognizer. *Proc. IEEE*, 73:1616–1624, 1985.

[21] F. Jelinek. Stochastic methods for context free grammars. In *Proc. NATO ASI Speech Understanding*, Cetraro, Italy, 1990.

[22] H. Kitano, H. Tomabechi, T. Mitamura, and H. Iida. A massively parallel model of speech-to-speech dialog translation. In *Proc. European Conf. on Speech Communication and Technology*, pages 198–201, 1989.

[23] D.H. Klatt. Review of the ARPA speech understanding project. *J. Acoust. Soc. America*, 62:1345–1366, 1977.

[24] S. Kunzmann. *Einsatz von Suchstrategien für Worthypothesen in Spracherkennungssystemen.* PhD thesis, Technische Fakultät, Universität Erlangen-Nürnberg, 1989.

[25] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 4(1):35–56, 1990.

[26] K.-F. Lee, H.-W. Hon, and R. Reddy. An overview of the SPHINX speech recognition system. *TransASSP*, 38:35–45, 1990.

[27] S.E. Levinson and L.R. Rabiner. A task-oriented conversational mode speech understanding system. *Bibliotheca Phonetica*, 12:149–196, 1985.

[28] B.T. Lowerre. *The HARPY Speech Recognition System.* PhD thesis, Dept. Comput. Sci., Carnegie-Mellon University, Pittsburgh, PA, 1976.

[29] T.B. Martin. One was to talk to computers. *IEEE Spectrum*, 14(5):35–39, 1977.

[30] M. Mast. *Entwicklung und Realisierung eines Dialogmoduls für ein System zum Verstehen kontinuierlich gesprochener Sprache.* Technical Report, Lehrstuhl für Informatik 5 (Mustererkennung), Universität Erlangen-Nürnberg, Erlangen, 1989.

[31] R. Moore, F. Pereira, and H. Murveit. Integrating speech and natural-language processing. In *Speech and Natural Language Workshop*, pages 243–247, Philadelphia, 1989.

[32] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 32:263–266, 1984.

[33] H. Niemann. Control strategies in image and speech understanding. In *Proc. GWAI*, pages 31–49, Springer, Berlin, 1983.

[34] H. Niemann. *Pattern Analysis and Understanding, 2. ed. Springer Series in Information Sciences 4.* Springer, Berlin, Heidelberg, 1990.

[35] H. Niemann, A. Brietzmann, U. Ehrlich, S. Posch, P. Regel, G. Sagerer, R. Salzbrunn, and G.Schukat-Talamazzini. A knowledge based speech understanding system. *Int. J. on Pattern Recognition and Artificial Intelligence*, 2:321–350, 1988.

[36] H. Niemann, A. Brietzmann, U. Ehrlich, and G. Sagerer. Representation of a continuous speech understanding and dialog system in a homogeneous semantic net architecture. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 30.6.1–30.6.4, Tokyo, Japan, 1986.

[37] H. Niemann, A. Brietzmann, R. Mühlfeld, P. Regel, and G. Schukat. The speech understanding and dialog system EVAR. In R. DeMori and C. Y. Suen, editors, *New Systems and Architectures for Automatic Speech Recognition and Synthesis*, Springer, Berlin, Heidelberg, New York, Tokyo, 1985.

[38] H. Niemann, M. Lang, and G. Sagerer, editors. *Recent Advances in Speech Understanding and Dialog Systems.* Volume 46 of *NATO ASI Series F*, Springer, Berlin, 1988.

[39] H. Niemann, G. Sagerer, and W. Eichhorn. Control strategies in a hierarchical knowledge structure. *Int. J. on Pattern Recognition and Artificial Intelligence*, 2:557–572, 1988.

[40] H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. ERNEST: A semantic network system for pattern understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:883–905, 1990.

[41] N.J. Nilsson. *Principles of Artificial Intelligence.* Springer, Berlin, Heidelberg, New York, 1982.

[42] L.R. Rabiner. Mathematical foundations of hidden markov models. In H. Niemann, M. Lang, and G. Sagerer, editors, *Recent Advances in Speech Understanding and Dialog Systems,*, pages 183–205, Springer, Berlin, 1988.

[43] P. Regel. *Akustisch-Phonetische Transkription für die automatische Spracherkennung. Fortschrittberichte VDI Reihe 10 Nr. 83*, VDI Verlag, Düsseldorf, 1988.

[44] G. Sagerer. *Automatisches Verstehen gesprochener Sprache.* Volume 74 of *Reihe Informatik*, BI Wissenschaftsverlag, Mannheim, 1990.

[45] E.G. Schukat-Talamazzini. *Generierung von Worthypothesen in kontinuierlicher Sprache.* Volume 141 of *Informatik Fachberichte*, Springer, Berlin, 1987.

[46] G. Schukat-Talamazzini and H. Niemann. Generating word hypotheses in continuous speech. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 1565–1568, Tokyo, 1986.

[47] S. Seneff. TINA: A probabilistic syntactic parser for speech understanding systems. In *Speech and Natural Language Workshop*, pages 168–178, Philadelphia, 1989.

[48] L. Tesniere. *Elementes des Syntaxe Structurale.* Klincksieck, Paris, 1966.

[49] W. von Hahn, W. Hoeppner, W. Jameson, and W. Wahlster. The anatomy of the natural language dialog system HAM-RPM. In L. Bolc, editor, *Natural Language Based Computer Systems*, chapter , pages 119–153, Carl Hanser, München, 1980.

[50] W.A. Woods. Optimal search strategies for speech understanding control. *Artificial Intelligence*, 18:295–326, 1982.

[51] S.J. Young and C.E. Proctor. The design and implementation of dialogue control in voice operated database inquiry systems. *Computer Speech & Language*, 3(4):329–353, 1989.

[52] S.R. Young and W.H. Ward. Towards habitable systems: use of world knowledge to dynamically constrain speech recognition. In *Proc. 2. Symp. Advanced Man-Machine Interface*, pages 30.1–30.12, Hawaii, 1988.

[53] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. The Voyager speech understanding system: a progress report. In *Proc. Second DARPA Speech and Natural Language Workshop*, Harwichport MA, USA, 1989.